COOPERATIVE BOOLEAN SYSTEMS WITH GENERICALLY LONG ATTRACTORS I

WINFRIED JUST AND MACIEJ MALICKI

Abstract. We study the class of cooperative Boolean networks whose only regulatory functions are COPY, binary AND, and binary OR. We prove that for all sufficiently large N and $c < 2$ there exist Boolean networks in this class that have an attractor of length $> c^N$ whose basin of attraction comprises an arbitrarily large fraction of the state space. The existence of such networks contrasts with results on various other types of dynamical systems that show nongenericity or absence of non-steady state attractors under the assumption of cooperativity.

1. Introduction

Understanding the role of feedback is crucial in the study of dynamical systems; see, *e.g.,* [23, 24] for relevant surveys. The absence of negative feedback loops tends to favor steady state attractors. For example, continuous flows without negative feedback loops are known as *monotone* systems. In these systems trajectories converge generically towards an equilibrium under mild regularity hypotheses; see *e.g.* [4, 12, 20]. Similarly, [18, 19] show that in Boolean networks with asynchronous updating negative feedback loops are necessary for the existence of attracting limit cycles.

Here we study a related question for Boolean networks with synchronous updating, which is the updating scheme originally proposed by Stuart Kauffman in his seminal papers [13, 14]. More specifically, we study the class of *cooperative* such Boolean networks which is defined by the total absence of negative interactions. Cooperative Boolean networks are the ones whose regulatory functions can be represented by compositions of AND, OR, and COPY functions, without the use of negations. This is a more restrictive notion than the absence of negative feedback loops, which are defined as containing an odd number of negative interactions.

In Boolean networks with synchronous updating negative feedback loops are not necessary for the existence of non-steady-state attractors, even exponentially long ones. It is shown in [3, 5, 21] that N-dimensional cooperative Boolean networks can have attractors of size up to $\binom{N}{\lfloor N/2 \rfloor} \sim \frac{2^N}{\sqrt{N}}$. Note, however, that this upper bound implies that no attractor in sufficiently high-dimensional cooperative Boolean networks can comprise a fixed fraction of the state space, whereas without the assumption of cooperativity even the whole state space can form a cyclic attractor of length 2^N. Moreover, in a randomly chosen N-dimensional cooperative Boolean network, with probability approaching one as $N \to \infty$, a steady-state

2000 *Mathematics Subject Classification.* 34C12, 39A33, 94C10.

Key words and phrases. Boolean networks, cooperative dynamical systems, exponentially long attractors, chaotic dynamics.

attractor will be reached from a randomly chosen initial condition in at most two steps [11].

This naturally raises the question whether there exist, for given constants p, c with $0 < p < 1 < c < 2$, cooperative Boolean networks of arbitrarily large dimension N for which the union of the basins of attraction of limit cycles of length $> 2^c$ comprises a fraction of at least p of the state space; a property that we call *p-c-chaos.* In this paper we answer this question in the affirmative. The Boolean systems that we construct use only regulatory functions with one or two inputs and are such that with probability $> p$ the trajectories of any two randomly chosen initial conditions will become equal in the long run, a property we call *p-coalescence.* Note that the latter implies that the basin of attraction of a single attractor of length $> 2^c$ comprises a fraction of $> p$ of the state space.

Before we formally state our theorems in Section 3, let us make some additional comments on their relation to the literature on the subject.

The dynamics of Boolean networks tends to fall either into the *ordered regime* that is characterized by short attractors, a large proportion of eventually frozen nodes, and low sensitivity to perturbations of initial conditions, or into the *chaotic regime* that is characterized by very long attractors, very few eventually frozen nodes, and high sensitivity to perturbations of initial conditions [15]. Note that p-c-chaos is a direct formalization of the first hallmark of chaos. It also implies, for suitable choices of p and c, a small proportion of eventually frozen nodes (see Proposition 3.2). On the other hand, p-coalescence implies low sensitivity of long-term behavior to perturbations of initial conditions. Thus the dynamics of the Boolean networks we construct here are extremely chaotic in one sense and highly ordered in another sense.

There exists a large body of literature on the *expected* dynamics of so-called *random Boolean networks (RBNs)* with certain restrictions on their connectivity and regulatory functions (see, *e.g.,* the surveys [2, 10, 15]). In these studies one considers a class $\mathcal{B}$ of Boolean networks with a given (usually uniform) probability distribution and tries to determine the expected values of the length of attractors, the proportion of eventually frozen nodes, or measures of sensitivity to initial conditions. For example, let $\mathcal{B}_{N,K}$ be the class of Kauffman's NK-networks, that is, Boolean networks with N variables and the sole restriction that each regulatory function can take at most K inputs. Then the expected dynamics of $\mathcal{B}_{N,K}$ becomes more chaotic as K increases from 1 to N in terms of all three features described above. In particular, p-c-chaos is generic in $\mathcal{B}_{N,N}$ when $p < 1$ and $c < \sqrt{2}$. This contrasts sharply with the situation for randomly chosen cooperative Boolean networks without any restrictions on the connectivity for which the above mentioned results of [11] show that p-c-chaos is about as non-generic as possible. In a similar vein, simulation studies [22] indicate that decreasing the amount of negative feedback in RBNs from $\mathcal{B}_{N,K}$ has the effect of decreasing the average lengths of the attractors.

While Boolean networks with fewer inputs per node tend to have on average shorter attractors, it is still possible to construct *quadratic* Boolean networks (that is, such that each node takes at most two inputs) with attractors whose length is a fixed fraction of the state space of very large dimensions [16, 17]. Cooperative *bi-quadratic* Boolean networks (that is, such that the number of in- and outputs per node is bounded from above by 2) can still have an attractor of size $> c^N$ for each $c < 2$ and all sufficiently large N [7, 8]. However, if in addition to the latter we require that at least a fixed fraction $\alpha > 0$ of all nodes take *exactly* two inputs, then the length of attractors is bounded from above by c_α^N for some $c_\alpha < 2$. In particular, $c_1 = 10^{1/4}$ and the bound is sharp [7, 8]. The constructions in [7, 8] do not appear to be adaptable to yielding p-c-chaotic cooperative Boolean networks for $p + c$ arbitrarily close to 3, which is achieved by the alternative construction described here. The Boolean networks in our construction are also bi-quadratic. While the latter property is not actually needed for the construction, it adds interest to our results in view of these bounds on attractor length.

2. Basic definitions

Let $[N]$ denote the set $\{1, \dots, N\}$ and let 2^N denote the set of all binary vectors with coordinates in $[N]$. An *N-dimensional Boolean system* $\mathbb{B}$ *(or Boolean network)* is a pair $\mathbb{B} = (2^N, f)$, where $f : 2^N \to 2^N$ is a Boolean function. While f uniquely determines $\mathbb{B}$, we will make a careful verbal distinction between a *Boolean network,* which is a pair, a *Boolean function* $f : 2^r \to 2^u$, and a *partial Boolean function* $f :\subseteq 2^r \to 2^u$.

For a Boolean network $\mathbb{B} = (2^N, f)$, the updating function f determines the successor state $\vec{s}(t + 1)$ of a Boolean state $\vec{s}(t) \in 2^N$ one time step later according to the formula

$$\vec{s}(t + 1) = f(\vec{s}(t)). \tag{1}$$

Note that $f = (f_1, \dots, f_N)$, where the components $f_i : 2^N \to \{0, 1\}$ give us the successor states for individual variables according to

$$s_i(t + 1) = f_i(\vec{s}(t)). \tag{2}$$

We will sometimes refer to f_i as the *regulatory function for variable i.* Since the state space 2^N is finite, the trajectory $\vec{s}(0), \vec{s}(1), \dots$ of an initial state $\vec{s}(0)$ must eventually revisit the same state and thus must either reach a *fixed point* $\vec{s}(t)$ (*i.e.* a state with $f(\vec{s}(t)) = \vec{s}(t)$) or a *cyclic attractor* of length $\leq 2^N$.

Now fix a Boolean network $\mathbb{B} = (2^N, f)$. For $\vec{s} = (s_1, \dots, s_N)$ and $i \in [N]$ let $\vec{s}^{\,j*} = (s_1, \dots, s_{j-1}, 1 - s_j, s_{j+1}, \dots, s_N)$ be the state obtained from $\vec{s}$ by a one-bit flip at position j. If for some $\vec{s}$ and j we have $f_i(\vec{s}) \neq f_i(\vec{s}^{\,j*})$, then we say that *variable j is an input of variable i* and *variable i is an output of variable j.* We call $\mathbb{B}$ *quadratic* if each variable takes at least one and at most two inputs and *bi-quadratic* if each variable takes one or two inputs and has at most two outputs. A partial Boolean function f is *cooperative* if $\vec{s} \leq \vec{s}^{\,+}$ implies $f(\vec{s}) \leq f(\vec{s}^{\,+})$, where $\leq$ denotes the coordinatewise partial order of Boolean vectors. By considering its conjunctive or disjunctive normal form, it is easy to see that a Boolean function is cooperative iff it can be represented as a composition of the functions COPY, AND, OR and Boolean constants. Note that $\mathbb{B}$ is quadratic and cooperative iff each regulatory function is either COPY, binary OR, or binary AND. We specifically exclude Boolean constants in this definition since our construction works without their help, which leads to a slightly stronger result.

The following fact will be useful in some of our constructions.

Proposition 2.1. *Let $f :\subseteq 2^r \to 2^u$ be a partial Boolean function.*

(a) If the domain of f consists of vectors that are pairwise incomparable in the coordinatewise partial order, then f is cooperative.

(b) If f is cooperative, then f can be extended to a cooperative total Boolean function $f^* : 2^r \to 2^u$.

Proof. Point (a) follows immediately from the definition of cooperativity. For the proof of (b) we can define the i-th coordinate $f^*(\vec{s})_i$ of $f^*(\vec{s})$ as the minimum of $f(\vec{s}^{\,+})_i$ over $\vec{s}^{\,+}$ in the domain of f with $\vec{s} \leq \vec{s}^{\,+}$. □

For $0 < p < 1 < c < 2$ we say that a Boolean system $\mathbb{B}$ is *c-chaotic* if it contains an attractor of length $> c^N$, and we say that $\mathbb{B}$ is *p-c-chaotic* if attractors of length $> c^N$ will be reached with probability $> p$, that is, if a proportion of $> p$ of all initial states belongs to basins of attraction of attractors of length $> c^N$.

A variable s_i is *eventually frozen* along a trajectory if the value $s_i(t)$ is fixed for all $\vec{s}(t)$ in the attractor. We say that $\mathbb{B}$ is *p-fluid* if with probability $> p$ the attractor of a randomly chosen initial state has a proportion of less than $1 - p$ eventually frozen variables.

Two initial states $\vec{s}^{\,0}(0), \vec{s}^{\,1}(0)$ *coalesce* if there exists a time $t \geq 0$ such that $\vec{s}^{\,0}(t) = \vec{s}^{\,1}(t)$. We call $\mathbb{B}$ *p-coalescent* if any two randomly and independently chosen initial conditions will coalesce with probability $> p$.

3. Statement of main result

We will prove the following result that strengthens Theorem 1(i) of [7].

Theorem 3.1. *Given any $0 < p < 1 < c < 2$, for all sufficiently large N there exist p-c-chaotic, p-coalescent, N-dimensional bi-quadratic cooperative Boolean networks.*

Let us briefly address the second hallmark of chaotic dynamics, few eventually frozen nodes, that was formalized above as p-fluidity. In [7], some additional effort was required to ensure that the c-chaotic systems constructed there are also p-fluid. This is not necessary here, since for sufficiently large c the property of p-c-chaos already implies p-fluidity.

Proposition 3.2. *Let $0 < p < 1$. Then there exists $c_p < 2$ such that for every c with $c_p < c < 2$ every p-c-chaotic Boolean system is p-fluid.*

Proof. Let p be as in the assumption. Let $c_p = 2^p$, and let $c_p < c < 2$. Consider an initial condition in the basin of attraction of an orbit of size $> c^N$. If a proportion of at least $1 - p$ of nodes are eventually frozen along the trajectory of this initial condition, then the attractor can have length at most $2^{pN} = c_p^N$, which contradicts the conditions on c. However, p-c-chaos implies that a proportion of more than p initial condition belongs to basins of attractions of periodic orbits of length $> c^N$, and thus implies p-fluidity. □

4. Some Terminology

Let us begin by introducing some terminology that will be used throughout the proof of Theorem 3.1. The set of positive integers will be denoted by $\mathbb{N}$ and $\log n$ denotes logarithm in base 2.

We generalize the notion $s_i(t)$ to $s_X(t)$, where X is a subset of the set of all variables. Thus $s_X(t) \in 2^X$ is the vector with coordinates $s_i(t)$ for $i \in X$.

If τ is a positive integer, and f^τ denotes the τ-th iteration of the updating function f of a given Boolean system, then

$$\vec{s}(t+\tau) = f^\tau(\vec{s}(t)). \tag{3}$$

In the Boolean systems we are going to construct, we will often have subsets X, Y of the variables so that the values of $s_i(t+\tau)$ for $i \in Y$ will depend only on $s_X(t)$. Thus we will construct Boolean systems that satisfy the following for some function $g : 2^X \to 2^Y$ and a fixed τ

$$s_Y(t+\tau) = f^\tau(\vec{s}(t)) \restriction Y = g(s_X(t)). \tag{4}$$

We will refer to the property expressed by (4) by writing that f *computes* g *on input* X *and writes the output to* Y *after* τ *steps.*

The function g will sometimes be given a verbal or formal description rather than be expressed by a separate symbol.

If $(r(t) : \ t \in \mathbb{N})$ is a given sequence of Boolean vectors in 2^Y, then we will say that f *writes* $r(t)$ *to* Y *with probability* $> q$ *for* $t \geq \tau$ if for a proportion of $> q$ of all initial conditions $\vec{s}(0) \in 2^N$ we have:

$$\forall t \geq \tau \ \ s_Y(t) = f^t(\vec{s}(0)) \restriction Y = r(t). \tag{5}$$

5. Outline of the proof of Theorem 3.1

Equipped with the terminology introduced in the previous section we are now ready to outline the proof of Theorem 3.1. More precisely, we will actually prove Theorem 3.1 in this section with the exception of a number of technical lemmas, some formal definitions, and a brief technical description in Section 9 of how the components of our construction fit together.

Let $0 < p < 1 < c < 2$ be as in the assumptions of Theorem 3.1.

The proof of our theorem boils down to constructing, for sufficiently large N, a suitable updating function f for a Boolean systems $\mathbb{B} = (2^N, f)$. We need to assure that f is cooperative, bi-quadratic, and works as expected, with probability $> p$, for randomly chosen (pairs of) initial conditions.

The set of Boolean variables $[N]$ will contain pairwise disjoint sets X_i, indexed by $i \in I$, where $I = \{0, 1, \dots, |I| - 1\}$. With probability $> p$ the following will hold for all times t of the form $t = t_0 + k|I|$, where t_0 is some fixed time and k is a nonnegative integer: With the possible exception of indices i in a small subset $Q \subset I$, each vector $s_{X_i}(t)$ will code an integer $v_i(t) \in \{0, \dots, n-1\}$ for some suitable value of n that depends on N. Moreover, again with the possible exception of $i \in Q$, the function f computes addition of 1 modulo $n - i$ on input X_i and writes the output to X_i after $|I|$ steps. Formally, the latter means that for times t as above

$$\forall\, i \in I \backslash Q \quad v_i(t + |I|) = v_i(t) + 1 \ mod \ (n - i). \tag{6}$$

Let $\mathcal{X} = \{X_i : i \in I\}$ and let $\mathcal{V} = \{V_i : i \in I\}$ be the functions that decode integers v_i from certain vectors s_{X_i}. We will also let $X = \bigcup_{i \in I} X_i$ and $Y = [N] \backslash X$. A structure $\mathcal{M} = (\mathbb{B}, p, I, Q, n, \mathcal{X}, Y, \mathcal{V}, t_0)$ will be called an *M-system* if it has the properties described above.

In the next section we will define a type of structure that we call an *MM-system.* Essentially, an MM-system is an M-system with several additional parameters and a number of size restrictions on how $N, n, |I|, |Q|, |X|, |Y|$ are related to each other. A major concern here is that we need to assure that $|Y|$ is small relative to $|X|$ and that $|Q|$ is sufficiently small relative to $|I|$. One of these additional parameters will be c; the other parameters will allow us to control the size restrictions mentioned above. We will use notations like $\mathcal{M}(\mathbb{B}, p, c)$ for MM-systems or M-systems with only the immediately relevant parameters shown. The implicit understanding will be that the remaining parameters are as specified in Definitions 6.1 and 6.2 of Section 6, where we will prove that for MM-systems $\mathcal{M}(\mathbb{B}, p, c)$ of sufficiently large dimensions the Boolean system $\mathbb{B}$ is p-c-chaotic (Lemma 6.3).

Now the proof of Theorem 3.1 boils down to proving Lemma 6.4 of Section 6, that is, to constructing an MM-system such that the updating function f is biquadratic and cooperative and such that with probability $> \sqrt{p}$ the trajectory of a randomly chosen initial state will reach a specified state at time t_0. Thus with probability $> p$ the trajectories of two randomly chosen initial states will coalesce at some time $t \leq t_0$.

We will deal with the size restrictions in the definition of a MM-system by showing that all functions that f needs to compute are what we call *L-functions.* We will formally define L-functions in Section 7; at the moment it suffices to say that L-functions are Boolean functions that can be implemented by cooperative biquadratic Boolean input-output systems of relatively low depth without adding too many variables. The restriction on the depth means that all necessary computations will require, for sufficiently large n, only $\tau \ll |I|$ steps, so that the output can be written to the required variables before it will interfere with other computations.

Let us conceptualize the collection of all X_i's as a circular data tape, with f simply copying the vector $s_{X_{i+1}}(t)$ to $s_{X_i}(t+1)$ for most i's, and also copying $s_{X_0}(t)$ to $s_{X_{|I|-1}}(t+1)$.

For $t \in \mathbb{N}$ let $t^* = t \textit{ mod } |I|$. We single out some $i_1, i_2 \in I$ with $i_2 = i_1 - \tau_1$ and construct f in such a way that f computes $v_{i_1} + 1$ modulo $n - t^*$ in τ_1 steps on input X_{i_1} and writes the output of this operation to X_{i_2}.

Let us for a moment assume this can be done and show how it implies (6) for all times $t \geq t_0$ under suitable assumptions on t_0 and $\vec{s}(t_0)$. Specifically, let us assume that $t_0 = i_1$ and that all vectors $s_{X_i}(t_0)$ are coding, with the possible exceptions of $i \in Q$. Then f computes $v_{i_1}(t_0) + 1$ modulo $n - i_1$ on input X_{i_1} and writes the output of this operation to X_{i_2} after τ_1 steps. In the next step we will have $v_{i_1}(t_0 + 1) = v_{i_1+1}(t_0)$, and thus f computes $v_{i_1+1}(t_0) + 1$ modulo $n - (i_1 + 1)$ on input X_{i_1} and writes the output of this operation to X_{i_2} after τ_1 steps. And so on. After $|I|$ time steps, the data tape will have come round circle, and we will have $v_i(t_0 + |I|) = v_i(t_0) + 1 \textit{ mod } (n - i)$ for all $i \in I$, and (6) follows by induction for all $t \geq t_0$.

The alert reader will have noticed that the variables in $X_{i_2+1} \cup X_{i_2+2} \cup \cdots \cup X_{i_1-1}$ are really not needed, since $s_{X_{i_2}}(t+1)$ will *not* be a copy of $s_{X_{i_2+1}}(t)$. We keep these variables here for the purpose of giving a uniform and more easily readable description of our construction. A similar remark applies to the variables $X_{i_1+1} \cup X_{i_1+2} \cup \cdots \cup X_{i_0-1}$, where i_0 will be defined shortly.

In order to make the idea of f computing $v_{i_1} + 1$ modulo $n - t^*$ work, we will need a counter for tracking t^*. The easiest way of implementing the counter is as a set of variables R so that $s_R(t)$ will be a code for $t^* = t \ mod\ |I|$.

Lemma 5.1. *There exists an L-function F_1 that computes $v_{i_1} + 1$ modulo $n - i$ in τ_1 steps on input (X_{i_1}, R) and writes its output to X_{i_2} if given input vectors $(s_{X_{i_1}}, s_R)$ such that $s_{X_{i_1}}$ codes the integer v_{i_1} and s_R codes the integer i.*

We will need to assure that at time t_0 the system will reach a state where $s_R(t) = b(t)$ for all times $t \geq t_0$ and the $b(t)$'s are specified Boolean vectors that make the construction work. This cannot be achieved for all possible initial conditions, only with probability arbitrarily close to one.

In Section 7 we will also define a notion of *L-sequences.* Essentially, these are periodic sequences of Boolean vectors, for which there are cooperative bi-quadratic Boolean input-output systems B that return this sequence with probability arbitrarily close to one, for all sufficiently large times. As for L-functions, there are restrictions on the depth and number of internal variables in B so that we can incorporate B into an MM-system. Moreover, we require that the vector of internal variables of B will eventually assume a sequence of specified states $h^*(t)$, for all times $t \geq t_0$, also with probability arbitrarily close to one.

Lemma 5.2. *Any specified sequence $(b(t) : t \in \mathbb{N})$ of values of s_R such that $b(t + |I|) = b(t)$ for all $t \in \mathbb{N}$ is an L-sequence.*

We also need to assure that the function F_1 of Lemma 5.1 receives input vectors s_{X_i} that do code for integers. Unfortunately, with probability close to 1, the vectors $s_{X_i}(0)$ in the initial state will *not* be coding any integers. In fact, our Lemma 9.1 shows that with probability arbitrarily close to 1 as $N \to \infty$, each of the vectors $s_{X_i}(0)$ will be belong to a set of *crude vectors* that we will rigorously define in Section 8.

All crude vectors are incomparable with all coding vectors in 2^{X_i}. The latter property turns out to be actually quite useful for our purposes. It allows us to incorporate a Boolean circuit into our definition of f so that it takes as inputs $(s_{X_{i_0}}, s_R)$ and writes an identical copy of the input $s_{X_{i_0}}$ to $s_{X_{i_1}}$ if the input consists of a pair of coding vectors, but outputs a fixed coding vector if at least one of the input vectors is crude. The circuit takes τ_3 time steps for its calculations, and we let $i_0 = i_1 + \tau_3$. Formally:

Lemma 5.3. *There exists an L-function F_3 on inputs (X_{i_0}, R) that writes its output to X_{i_1} and returns a specified coding vector x^* on all inputs with the property that $s_{X_{i_0}}$ or s_R is crude and that returns an identical copy of $s_{X_{i_0}}$ on all pairs of input vectors that code integers.*

Now let $0 < q_1, q_2 < 1$ be probabilities such that $q_1 + q_2 - 1 > \sqrt{p}$, that is, such that if events E, F occur with probabilities $> q_1$ and $> q_2$ respectively, then event $E \cap F$ occurs with probability $> \sqrt{p}$. Let N be sufficiently large so that with probability $> q_1$ the Boolean input-output system B_2 that returns the sequence of $b(t)$'s for all times $t \geq \tau_2$, where $\tau_2 \ll |I|$, behaves as desired and with probability $> q_2$ all initial vectors $s_{X_i}(0)$ are crude. We will construct the updating function f so that it is cooperative, bi-quadratic, computes the functions F_1, F_3 in the sense of (4) with inputs and outputs as specified by Lemmas 5.1 and 5.3, with appropriate values of $b(t)$ and, with probability $> q_1$, writes these values $b(t)$ to $s_R(t)$ for all times $t \geq \tau_2$. The formal proof of Lemma 6.4 in the remainder of this paper shows that it is possible to construct such $f : 2^N \to 2^N$ for all sufficiently large dimensions N in such a way that $|I|$ exceeds the combined number of time steps of all necessary computations.

With probability $> \sqrt{p}$, this presents us with the following situation after $t_2 := \max\{\tau_2, \tau_3\}$ time steps: The inputs $s_{X_{i_1}}, s_R$ of F_1 are pairs of codes for integers as needed for the computation of $v_i + 1$ modulo $n - i$, since $s_R = b(t_2)$ is the desired value of the counter and $s_{X_{i_1}}$ is the value of F_3 for an input with a crude coordinate $X_{i_0}(0)$. The analogous property remains true at subsequent time steps. Now let us move the system forward to time $t_1 := t_2 + \tau_1$. At this time, the vectors $s_{X_{i_2}}, s_{X_{i_2-1}}, \dots, s_{X_{i_2-t_1+1}}$ will be copies of output of f for values of the input vectors on X_{i_1}, R that with high probability are not pairs of codes for integers. We don't have much control over these vectors; it is not at all clear whether they are coding or crude. So we cannot automatically assume that they will be turned into coding vectors once they will have migrated to position i_0. This is were Q comes in: We let $Q = \{i_2, i_2 - 1, \dots, i_2 - t_1 + 1\}$. One may consider Q a set of "possibly corrupted memory locations." Our formulation of (6) allows us to disregard these memory locations. We will tag these locations by specifying a fixed crude vector $b(t)$ for all those times t such that F_3 takes an input vector $s_{X_{i_0}}$ at time t which is a copy of some $s_{X_i}(t_1 + m|I|)$ with $i \in Q$, where m is a nonnegative integer. The crude input coordinate at these times t will ensure that F_3 writes the specified coding vector x^* to X_{i_1} at the corresponding times $t + \tau_3$.

Now consider $t_0 = t_1 + |I|$. With probability $> \sqrt{p}$ the following holds: All the vectors $s_{X_i}(t_0)$ with $i \notin Q$ will be coding. For all $t \geq t_0 > t_1$, by Lemma 5.2, the vectors $s_R(t)$ can be required to hold codes $b(t)$ for integers so that the function F_1 will add 1 mod $(n - t^*)$ to $v_{i_1}(t)$ and write the output to X_{i_2} at time $t + \tau_1$, where $t^* = t \ mod \ |I|$, unless $s_{X_{i_1}}(t)$ is a copy of some $s_{X_i}(t_1 + m|I|)$ with $i \in Q$. The function F_3 will simply copy $s_{X_{i_0}}(t)$ to $s_{X_{i_1}}(t + \tau_3)$ unless the associated $b(t)$ indicates that $s_{X_{i_0}}(t)$ originates from some $s_{X_i}(t_1 + m|I|)$ with $i \in Q$. Thus for $t = t_0$ property (6) will hold, and by induction, the same will be true for all times t of the form $t = t_0 + k|I|$ for some nonnegative integer k. Since $q_1 + q_2 - 1 \geq p$, the latter implies that $\mathbb{B}$ thus constructed can be part of an MM-system, and it follows from Lemma 6.3 that $\mathbb{B}$ is p-c-chaotic.

A slight technical difficulty arises from the need for F_1 and F_3 to receive the signal of a corrupted memory location given by a crude $b(t)$ at different times; we will show in Section 9 how to implement the necessary phase shift.

For the proof of p-coalescence, we will need in addition that with probability $> \sqrt{p}$ at time t_0 all variables in the system will assume values specified by a fixed vector $\vec{s}^+$. The argument for this works for randomly chosen initial states with probability $> q_1 + q_2 - 1 > \sqrt{p}$ and goes as follows. Our set of variables is $X \cup Y$, where Y comprises the internal variables of the Boolean input-output systems B_1, B_3 that implement F_1, F_3, and B_2 that returns the sequence of $b(t)$'s, plus some dummy variables. We may choose τ_2 such that already starting from time $t_2 < t_1$, with probability $> q_1$ the vector of all internal variables of B_2 will assume a fixed sequence of values $h^*(t)$ for all times $t \geq t_0$ by our definition of an L-sequence. With probability $> q_2$, all initial values s_{X_i} will be crude; as long as F_3 receives an unaltered copy of these as its first input, it will turn it into the specified coding vector x^*. Similarly, the vectors $s_{X_i}(t_1)$ for $i \in Q$ will be copied to inputs of F_3 and converted by F_3 into the value x^* after τ_3 time steps. All these operations will be completed before time t_0. Thus at time t_0, all vectors s_{X_i} will have values specified by the fixed sequence of $b(t)$'s and the output of F_1 on input vectors $(x^*, b(t))$. Moreover, the states of the internal variables of the Boolean circuit B_1 at time t_0 will be determined by the sequence of the inputs between times $t_0 - \tau_1$ and $t_0 - 1$, which is fixed. A similar observation holds for the the states of the internal variables of the Boolean circuit B_3 at time t_0. The value of any other variable at time t will simply be a copy of the value of some output variable of B_2 or of some variable in X at time $t - \tau$, where $\tau < \tau_3$. So we will get coalescence on those variables as well, and for a proportion of $> \sqrt{p}$ of initial states, $\vec{s}(t_0)$ will be the same state $\vec{s}^{\,+}$.

6. MM-systems

Let us begin by formally stating the definitions of an M-system and an MM-system that were described somewhat informally in the previous section.

Definition 6.1. *An N-dimensional M-system will be a mathematical object $\mathcal{M} = (\mathbb{B}, p, I, Q, n, \mathcal{X}, Y, \mathcal{V}, t_0)$ such that*

- *$\mathbb{B}$ is an N-dimensional Boolean system.*
- *p is a probability such that $0 < p < 1$.*
- *$I = \{0, 1, \ldots, |I| - 1\}$ is a set of consecutive nonnegative integers.*
- *$Q \subset I$.*
- *n is a positive integer such that $\log(n)$ is an integer.*
- *$\mathcal{X} = \{X_i : \ i \in I\}$ is a family of pairwise disjoint subsets of the set $[N]$ of Boolean variables of $\mathbb{B}$.*
- *$Y = [N] \backslash \bigcup \mathcal{X}$.*
- *$\mathcal{V} = \{V_i : \ i \in I\}$ is a family of partial functions such that V_i maps a subset of 2^{X_i} onto $\{0, \ldots, n-1\}$.*
- *t_0 is a positive integer.*
- *If s_{X_i} is in the domain of V_i, then $V_i(x_i)$ will be denoted by v_i.*
- *For a proportion of $> p$ of all initial conditions the following will hold for all t of the form $t = t_0 + k|I|$, where k is a nonnegative integer and t_0 is a fixed time, and all $i \in I \backslash Q$:*

  s_{X_i} is in the domain of V_i and

$$v_i(t + |I|) = v_i(t) + 1 \ mod(n - i + 1). \tag{7}$$

As mentioned in the previous section, an MM-system is an M-system with certain additional parameters and size restrictions that will imply p-c-chaos in $\mathbb{B}$.

Definition 6.2. *An N-dimensional MM-system is a mathematical object $\mathcal{M} = (\mathcal{M}^-, c, \varepsilon, \delta, \beta, \nu)$ with the following properties:*

- *$\mathcal{M}^- = (\mathbb{B}, p, I, Q, n, \mathcal{X}, Y, \mathcal{V}, t_0)$ is an N-dimensional M-system.*
- *$I = \{0, 1, \ldots, \beta\log(n) - 1\}$.*
- *$c, \varepsilon, \delta, \beta, \nu$ are constants such that $1 < 2 < c$; $\beta, (1 + \varepsilon)\log(n)$ are positive integers, $\varepsilon, \delta > 0$, $\beta > \nu \geq 0$, and*

$$\log(c)(1 + \varepsilon + \delta) < 1. \tag{8}$$

$$\forall i \in I \ |X_i| = (1 + \varepsilon)\log(n). \tag{9}$$

$$|Y| \leq \left(\beta\delta - \frac{\nu}{\log c}\right)\log^2(n). \tag{10}$$

$$|Q| \leq \nu\log(n). \tag{11}$$

Assume that $\mathcal{M}$ is an N-dimensional MM-system. By (9) and the choice of I we have

$$|X| = \beta(1+\varepsilon)\log^2(n). \tag{12}$$

Moreover, by (12) and (10) we will have

$$\beta(1+\varepsilon)\log^2(n) \leq N \leq \left(\beta(1+\varepsilon+\delta) - \frac{\nu}{\log c}\right)\log^2(n). \tag{13}$$

Note also that by our choice of I, equation (7) becomes

$$v_i(t+\beta\log(n)) = v_i(t) + 1 \; mod(n-i+1). \tag{14}$$

It will often be more convenient to use (7) written in the form (14).

Lemma 6.3. *Let p, c be constants such that $0 < p < 1 < c < 2$. Then for all sufficiently large N and every N-dimensional MM-system $\mathcal{M}(\mathbb{B}, p, c)$, the Boolean system $\mathbb{B}$ is p-c-chaotic.*

Proof. Let $\mathcal{M}(\mathbb{B}, p, c)$ be an N-dimensional MM-system with N (and hence n) sufficiently large.

Let $\vec{s}(0)$ be a randomly chosen initial condition and let A be the attractor that the trajectory of $\vec{s}(0)$ eventually reaches. With probability $> p$, condition (14) will hold for all t of the form $t = t_0 + k|I|$ such that $\vec{s}(t) \in A$. If this is the case, then we have $\vec{s}(t_1\beta\log(n)) \neq \vec{s}(t_2\beta\log(n))$ whenever $\vec{s}(t_1\beta\log(n)) \in A$ and $t_1 < t_2$ are such that $|t_2 - t_1| < n^{-|Q|}LCM(n, n-1, \dots, n-\beta log(n)+1)$. Thus

$$|A| \geq \frac{LCM(n, n-1, \dots, n-\beta log(n)+1)}{n^{\nu\log(n)}}. \tag{15}$$

It is known that

$$\mathrm{LCM}(n(n-1)\dots(n-k)) \geq \frac{(n-k)^k}{k!}$$

for any $n > k \in \mathbb{N}$ (see Theorem 4 of [6]). By Stirling's formula,

$$\frac{(n-k)^k}{k!} \approx \frac{(n-k)^k e^k\sqrt{2\pi k}}{k^k} \geq \frac{(n-k)^k}{k^k}\frac{e^k\sqrt{2\pi k}}{k^k} \geq \frac{n^k}{k^{2k}} \geq 2^{k\log(n)-2k\log(k)},$$

so, for $k = \beta\log(n)$, we have

$$LCM(n(n-1)\dots(n-\beta\log(n))) \geq 2^{\beta\log^2(n)-2\beta\log(n)\log(\beta\log(n))}. \tag{16}$$

From (15) and (16) we get

$$|A| \geq 2^{(\beta-\nu)\log^2(n)-2\beta\log(n)\log(\beta\log(n))}. \tag{17}$$

On the other hand, by (13) we have

$$c^N < 2^{(\log(c)(1+\varepsilon+\delta)\beta-\nu)\log^2(n)}. \tag{18}$$

For fixed β, ν, the term $2\beta\log(n)\log(\beta\log(n))$ becomes negligible as long as n is sufficiently large. Thus, by (8) the right-hand side of (17) will exceed the the right-hand side of (18), and Lemma 6.3 follows. □

The following lemma implies Theorem 3.1. In view of Lemma 6.3, its first part implies p-c-chaos, and its last sentence implies p-coalescence.

Lemma 6.4. *Let p, c be constants such that $0 < p < 1 < c < 2$. Then there exists a positive integer $N_{p,c}$ such that for all $N > N_{p,c}$ there exists an N-dimensional MM-system $\mathcal{M}(\mathbb{B}, p, c, t_0)$ such that $\mathbb{B}$ is bi-quadratic and cooperative. Moreover, this system can be constructed in such a way that there exists a state $\vec{s}^{\,+}$ so that for a randomly chosen initial condition $\vec{s}(0)$ we will have $\vec{s}(t_0) = \vec{s}^{\,+}$ with probability $> \sqrt{p}$.*

7. Boolean input-output systems

A *Boolean input-output system* is a hierarchical arrangement of binary variables, with the input variables constituting the lowest level, the output variables the highest level and each variable (except for the ones at the input level) taking input from one or several variables at some level(s) other than the output level and updating their current state according to an AND, OR, NOT gate, or simply copying its single input.

A *Boolean circuit* is a Boolean input-output system in which all the variables (except for the ones at the input level) take input from one or several variables at the next lower level.

The *depth* $d(B)$ of a Boolean input-output system B is the number of its levels, not counting the lowest (input) level. An *internal variable* of a Boolean input-output system is a variable that is neither an input variable nor an output variable. We will use the notation $B(D, H, R)$ to indicate the sets of variables of a Boolean input-output system B. In this notation, the sets D, H, R are pairwise disjoint, the domain D contains the input variables, the range R denotes the set of output variables, and H (for *hidden*) denotes the set of internal variables. Notice that the depth of $B(D, H, R)$ is 1 iff $H = \emptyset$.

We will call a Boolean input-output system $B = B(D, H, R)$ *cooperative* if it does not use NOT gates and *quadratic* if it uses only binary or unary gates. A quadratic system is *bi-quadratic* if each variable in $D \cup H$ serves as input for at most two variables in the system. We say that B is *monic* if every gate takes only one input. Note that this definition allows B to be simultaneously monic and bi-quadratic.

Let $B(D, H, R)$ be a Boolean circuit of depth d and let $g :\subseteq 2^D \to 2^R$ be a (partial) Boolean function. We say that B *calculates g (in d steps)* if for every input $\vec{s}$ in the domain of g after d updating steps the value of the output vector of B will be $g(\vec{s})$.

Example 1. *Let D, R be such that $|D| = |R|$ and let $\chi : D \to R$ be a bijection. Define $id : 2^D \to 2^R$ as the function that extends χ to binary vectors in the obvious* *way. Then there exists a cooperative bi-quadratic Boolean circuit $B_{id} = B_{id}(D, \emptyset, R)$ of depth 1 that calculates the function id.*

We can construct B_{id} in such a way that for $i \in D$ the value of $s_i(t)$ will simply be copied to $s_{\chi(i)}(t+1)$ by a monic regulatory function. This makes B_{id} cooperative, bi-quadratic, and monic.

Note that since in a Boolean circuit variables at each level take input only from the variables at the next lower level, the output of a Boolean circuit of depth d (after d steps) will not be influenced by the initial state of its internal variables. In contrast, the output of a Boolean input-output system after d time steps may also depend on the initial states of its internal variables.

Let $B(D, H, R)$ be a Boolean input-output system of depth d and let $(s_R(t) : t \in \mathbb{N})$ be a sequence of Boolean vectors in 2^R. We say that B *returns* $s_R(t)$ *with probability* $> q$ *for all* $t \geq d(B)$ if for a proportion of $> q$ of all initial states of the internal variables of B the output sequence will be as specified for all times $t \geq d(B)$, for any sequence $(s_D(t) : t \in \mathbb{N})$ of values of the input variables. This makes the input variables redundant, but for some constructions in the follow-up paper [9] we will need to include nonempty sets D of input variables. For this reason we give the definition here in its full generality rather than restricting it to the case $D = \emptyset$, which is the only one needed here.

Boolean circuits $B_1(D_1, H_1, R_1)$ and $B_2(D_2, H_2, R_2)$ can be concatenated in the obvious way as long as $R_2 = D_1$. The concatenation $B = B_1 \circ B_2$ will have input variables D_2, output variables R_1, internal variables $H = H_2 \cup R_2 \cup H_1$, and its depth will satisfy $d(B) = d(B_1) + d(B_2)$. Moreover, if B_1, B_2 are Boolean circuits that calculate g_1, g_2 respectively, then B will calculate $g_1 \circ g_2$. One can also convert Boolean input-output systems into parts of an updating function f of a Boolean network by re-interpreting the logical gates as updating functions for the relevant variables. The computations performed by the Boolean input-output systems then guarantee the corresponding computations of the updating function f in the sense of (4).

Now we are ready to define L-functions and L-sequences. Since we are interested in constructing Boolean systems of all sufficiently large dimensions, strictly speaking, these objects are really families of functions or sequences that contain one representative for each desired dimension N. In order to keep the terminology in our already rather technical proof reasonably manageable, we suppress reference to the whole family whenever this seems to make our arguments more transparent. The size of the domains and ranges of individual L-functions (L-sequences) will be controlled by the parameter n in the definition of an MM-system. In Section 9 we will define a set $\mathbb{M}$ of *suitable* n, which is, a set of all $n \in \mathbb{N}$ that could appear as parameter in at least one MM-system of interest. Our families of L-functions and L-sequences will be indexed by this set.

Definition 7.1. *Let $\mathcal{F} = (F^n : n \in \mathbb{M})$ be a family of functions with $F^n : D(n) \to R(n)$. We say that $\mathcal{F}$ consists of* L-functions *if there exist $\gamma > 0$ such that for every $n \in \mathbb{M}$ there exists a cooperative bi-quadratic Boolean circuit with $\leq \gamma(log(n))^2$ variables of depth $\leq \gamma \log(n)$ that calculates F^n.*

Definition 7.2. *Let $\mathcal{S} = (S(n) : n \in \mathbb{M})$ be a family of sequences with $S(n) = (s^n_R(t) : t \in \mathbb{N})$. We say that $\mathcal{S}$ consists of* L-sequences *if for every given probability*

$q < 1$ *there exists* $\gamma > 0$ *such that for every* $n \in \mathbb{M}$ *there exists a cooperative bi-quadratic Boolean input-output system* $B = B(\emptyset, H, R)$ *with* $\leq \gamma(log(n))^2$ *variables of depth* $d \leq \gamma \log(n)$ *that* returns $s^n_R(t)$ with probability $> q$ for all $t \geq d(B)$, *and a fixed sequence of vectors* $h^*(t) \in 2^H$ *such that with probability* $> q$ *the equality* $s_H(t) = h^*(t)$ *will hold for all* $t \geq 2d(B)$.

Note that certain size restrictions on $D(n)$ and $R(n)$ will be necessary to prove that a given sequence $\mathcal{F}$ is an L-function. Similarly, certain size restrictions on $s^n_R(t)$ and the periods T_n will be needed to prove that a given $\mathcal{S}$ that consists of sequences $s^n_R(t)$ with periods T_n is an L-sequence. These size restrictions have not been spelled out explicitly in Section 5; they will be made explicit in Section 9, and will also be spelled out in Section 10, where we prove Lemmas 5.1, 5.2, and 5.3. Let us just mention here that for a periodic sequence to be an L-sequence it is actually sufficient that $|s^n_R|T_n$ grows sufficiently slowly relative to n.

8. Coding vectors and crude vectors

For an even positive integer k let C_k be the set of all Boolean vectors from 2^k such that exactly half of their coordinates are 1's (so the other half are 0's). Then $(C_k)^\ell$ is a set of pairwise incomparable vectors for every positive integer ℓ. For every $i \in I$ we will choose sets $\mathbf{C}_i \subset 2^{X_i}$ of coding vectors with $\mathbf{C}_i \subset (C_k)^\ell$ for some suitable ℓ. Of course, the sets of variables X_i are pairwise disjoint, so we cannot literally make each $\mathbf{C}_i$ a subset of $(C_k)^\ell$; formally we will need disjoint copies of $(C_k)^\ell$. However, adding an extra parameter (as in: $(C_k(i))^\ell$) appears to introduce only clutter and we will use the slightly informal notation for the sake of transparency. Similarly, we will require that if $b_i \in 2^{R_i}$ codes for an integer, then $b_i \in (C_k)^m$ for some suitable m. This assures that all our codes for integers will be pairwise incomparable and allows us to construct cooperative Boolean functions since every partial Boolean function that is defined on a set of pairwise incomparable Boolean vectors can be extended to a cooperative Boolean function (see Proposition 2.1).

On the other hand, we need to be able to define values $v_i \in \{0, \dots, n-1\}$ for $x_i \in \mathbf{C}_i$ so that we can achieve (7). Thus we want $\mathbf{C}_i$ to be sufficiently large so that there exists a bijection $V_i : \mathbf{C}_i \to \{0, \dots, n-1\}$. We will show next that this is always possible if we judiciously choose k and the parameter ε of our MM-system. The vectors x_i in the sets $\mathbf{C}_i$ will henceforth be called *coding vectors.*

Definition 8.1. *Let c be a constant with $1 < c < 2$ and let $\varepsilon > 0$. We say that the pair (k, ε) is* c-friendly *if*

$$\varepsilon \text{ is rational and } \frac{k}{1+\varepsilon} \text{ is an integer,} \tag{19}$$

$$\log(c)(1+\varepsilon) < 1, \quad \text{and} \tag{20}$$

$$|C_k| \geq 2^{k/(1+\varepsilon)}. \tag{21}$$

Lemma 8.2. *Suppose $1 < c < 2$. Then there exist a rational $\varepsilon = \varepsilon(c) > 0$ and a positive even integer $k = k(c)$ such that the pair (k, ε) is c-friendly.*

Proof. Fix any rational $\varepsilon(c) > 0$ such that $\log(c)(1+\varepsilon(c)) < 1$. Note that $|C_k| = \binom{k}{k/2} \geq \frac{2^k}{k}$. Thus for sufficiently large even k we will have $|C_k| \geq 2^{k/(1+\varepsilon)}$. □

9. The proof of Lemma 6.4

Having defined all its ingredients, let us formally describe the construction of MM-systems for the proof of Lemma 6.4. Given $0 < p < 1 < c < 2$, pick ε and k such that the pair (k, ε) is c-friendly. An integer n will be called *suitable* if there exists an integer ℓ such that

$$(1+\varepsilon)\log(n) = k\ell \tag{22}$$

The set $\mathbb{M}$ in Definitions 7.1 and 7.2 will be the set of all suitable n.

Pick δ with $0 < \delta_0 < \delta < 1$ such that (8) holds. The latter is possible by (20). We will apply Lemma 5.2 to a periodic sequence with period T of vectors $s_R \in 2^R$ such that for all sufficiently large N:

$$|R|T \leq \delta_0 \log(n). \tag{23}$$

Let $\gamma_1, \gamma_3 > 0$ be constants that witness that the functions F_1, F_3 of Lemmas 5.1 and 5.3 are L-functions, and let γ_2 be a constant that witnesses that the sequence of values of the counter is an L-sequence whenever the size restriction (23) holds. Let $\gamma = \gamma_1 + \gamma_2 + \gamma_3 + 1$, and let $\nu = \gamma_1 + \gamma_2 + 2\gamma_3$.

Choose a sufficiently large positive integer β such that

$$\gamma \leq \left(\beta\delta_0 - \frac{\nu}{\log c}\right). \tag{24}$$

For sufficiently large N, choose a suitable n such that

$$\beta(1+\varepsilon+\delta_0)\log^2(n) \leq N \leq \left(\beta(1+\varepsilon+\delta) - \frac{\nu}{\log c}\right)\log^2(n). \tag{25}$$

This determines the set $I = \{0, 1, \dots, \beta\log(n) - 1\}$ and allows us to choose sets of variables X_i for $i \in I$ with $X = \bigcup_{i \in I} X_i$ such that (9) and (12) hold.

For each $i \in I$ we will identify a set of pairwise incomparable coding vectors $\mathbf{C}_i$ of 2^{X_i} that correspond to a subset of a copy of $(C_k)^\ell$ and a bijection $V_i : C_i \to \{0, \dots, n-1\}$ that computes the values $v_i(x_i)$ that are coded by $x_i \in \mathbf{C}_i$. The existence of $\mathbf{C}_i$ and V_i follows from (21) and the choice of n, ℓ. Now we can choose m as the smallest integer with $|(C_k)^m| \geq |I|$ and choose a set $\mathbf{C} \subseteq (C_k)^m$ of codes b_i for the integers $i \in I$. Finally, we let R be a set of size km; we will treat b_i as an element of 2^R. It follows from our choice of m that for some constant γ_R we have

$$|R| \leq \gamma_R \log(\log(n)). \tag{26}$$

The period of the counter will be $T = |I| = \beta\log(n)$. Thus T and $|R|$ depend on β, but regardless of the choice of β the estimate (23) will hold for sufficiently large N.

Let us fix i and write $X_i \cup R$ as a disjoint union of consecutive intervals Z_i^j of length k each, where $j < \ell + m$ and let z_i^j denote the respective truncations of $z_i \in 2^{X_i \cup R}$ to these intervals. Let us say that z_i is *crude* if there exist $j, J < \ell + m$

such that $z_i^j = \vec{0}$ and $z_i^J = \vec{1}$. Similarly $x_i \in 2^{X_i}$ is crude if we can find $j, J < \ell$ with this property. Note that all crude vectors are incomparable with all coding vectors.

Consider a randomly chosen initial state, and let E be the event that all $x_i(0)$ are crude.

Lemma 9.1. *Let k, ε, β be fixed. Then $P(E) \to 1$ as $n \to \infty$.*

Proof. For each i, the probability that $x_i(0)$ is not crude is $\leq 2(1-2^{-k})^\ell$. Thus the probability of the complement of E is $\leq 2\beta \log(n)(1-2^{-k})^\ell$. Since $\log(n) = \frac{k}{1+\varepsilon}\ell$ and k, ε, β are fixed, the result follows. □

Let q_1, q_2 be probabilities such that if events E, F occur with probabilities $> q_1$ and $> q_2$ respectively, then event $E \cap F$ occurs with probability $> \sqrt{p}$. In particular, we will assume that N is sufficiently large so that $P(E) > q_1$.

Let $i_0 = |I| - 1$ and choose two disjoint sets of variables R, R_c with $|R| = |R_c|$ as in (26). Choose a Boolean circuit $B_3 = B_3(X_{i_0} \cup R, H_3, X_{i_1})$ that witnesses that the function F_3 of Lemma 5.3 is an L-function, where $i_1 = i_0 - d(B_3)$. Then choose a Boolean circuit $B_1 = B_1(X_{i_1} \cup R_c, H_1, X_{i_2})$ that witnesses that the function F_1 of Lemma 5.1 is an L-function, where $i_2 = i_1 - d(B_1)$. By Definition 7.1 our choice of the input sets of the Boolean input-output systems implies in view of (26) that we will have $\tau_1 = d(B_1) \leq \gamma_1(\log(n) + \gamma_R \log(\log(n)))$ and $\tau_3 = d(B_3) \leq \gamma_3(\log(n) + \gamma_R \log(\log(n)))$. For sufficiently large n the terms $\gamma_R \log(\log(n))$ become negligible, and our choices of ν, β and I imply that $i_2 > 0$.

Choose a Boolean input-output system $B_2 = B_2(\emptyset, H_2, R)$ that witnesses that the sequence $b(t) \in 2^R$ of desired values of the counter is an L-sequence for the function that returns $b(t) \in 2^R$ as needed for the argument presented in Section 5. Make sure that the sets H_1, H_2, H_3, R, R_c, X are pairwise disjoint. These choices determine the regulatory functions for all variables in $B_1 \cup B_2 \cup B_3$, except for the variables in $X_{i_0} \cup R_c$.

For all $i \in I \backslash \{i_1, i_2\}$ define the regulatory functions for the variables in X_i so that they implement the circuit $B_{id}(X_{i+1}, \emptyset, X_i)$ of Example 1, where $|I| - 1 + 1$ is treated as zero.

Notice that for the proper working of B_3 it only matters whether the value s_R at a given time t is coding or crude, where the latter property signals that the input variables in X_{i_0} originate from a possibly corrupted memory location X_i with $i \in Q$. As already mentioned in Section 5, the circuit B_1 needs to receive the same signal $\tau_3 = d(B_3)$ time steps later. This is where R_c comes in: We make R_c the set of output variables of a Boolean circuit $B_4 = B_4(R, H_4, R_c)$ of depth τ_3 that is the concatenation of τ_3 copies of B_{id} and simply copies the values of the input variables in R to the output variables in R_c in τ_3 steps.

This describes the part of our construction that is needed for p-c-chaos in the resulting system $\mathbb{B}$. Let $\tau_2 = d(B_2)$. Note that as long as N is chosen sufficiently large, we can assume that with probability $> q_2$ for all times $t \geq \tau_2$ the values $s_R(t)$ will be the required $b(t)$'s. Thus with probability $> q_2$, at all times $t > \tau_2 + \tau_3$ the circuit B_1 will receive the required input values $b(t)$ on its input variables in R_c. Moreover, at all times $t > \tau_3$ the inputs $s_{X_{i_1}}$ of B_1 will be coding vectors. Thus with probability $> \sqrt{p}$ the circuit B_1 will write the desired output to X_{i_2} for all times $t > \tau_1 + \tau_2 + \tau_3$. It follows that the set Q of possibly corrupted memory locations has cardinality $\leq \tau_1 + \tau_2 + \tau_3$. Since $\tau_i \leq \gamma_i \log(n)$ by the definitions of L-functions and L-sequences, our choice of ν implies (11), and the choice of β implies that the block of corrupted memory locations does not extend all around the "data tape," that is, implies $0 \notin Q$. Now the argument presented at the end of Section 5 shows that condition (7) will hold with probability $> \sqrt{p} > p$ and that we get p-coalescence on the variables for which have already defined regulatory functions.

The part of the system we have constructed so far consists of variables in the sets $X = \bigcup_{i \in I} X_i$ and $Y^- := R_c \cup R \cup H_1 \cup H_2 \cup H_3 \cup H_4$ and regulatory functions for them. Also, since all Boolean input-output systems that we have used so far are cooperative and bi-quadratic, the part of the system that we have constructed so far inherits these properties.

By Definitions 7.1 and 7.2 and our choice of γ we have

$$|Y^-| \leq \gamma \log^2(n) \tag{27}$$

as long as n is sufficiently large so that $\gamma_R \log(\log(n)) \leq \log(n)$. We may need to add to Y^- a set of dummy variables so that the resulting set Y satisfies $|X|+|Y| = N$. By (25) and (12), the resulting Y will satisfy (10). We simply define the regulatory function for each dummy variable to be the function that copies a value from $X \backslash (X_{i_0} \cup X_{i_1})$. This retains cooperativity of the whole system. Since so far each variable in X serves as input to at most one other variable and since $|Y| < |X| - 2\log(n)$, we have enough different input variables at our disposal to assure that the resulting $\mathbb{B}$ will be bi-quadratic.

The resulting structure satisfies all conditions of an MM-system, which proves Lemma 6.4.

10. The proofs of Lemmas 5.1, 5.2, and 5.3.

10.1. Preliminary results.

Proposition 10.1. *Let $\Delta : \{0,1\} \to 2^Y$ be the function that produces $|Y|$ identical copies of a variable, defined by $(\Delta(x))_y = x$ for $y \in Y$. Then there exists a bi-quadratic cooperative Boolean circuit $B^{|Y|}_\Delta$ with $\leq 2|Y|$ internal variables that calculates Δ in $\leq \log(|Y|)$ steps.*

Proof. We consider only bi-quadratic systems, so at one step the content of a variable can be copied to two variables only. It means that we need $|Y|+|Y|/2+\ldots+1 \leq 2|Y|$ variables and $\leq \log(|Y|)$ steps. □

Proposition 10.2. *The functions $\wedge : 2^X \to \{0,1\}$ and $\vee : 2^X \to \{0,1\}$, defined by*

$$\wedge(x) = \wedge(x_1, \ldots, x_n) = x_1 \wedge \ldots \wedge x_n,$$

$$\vee(x) = \vee(x_1, \ldots, x_n) = x_1 \vee \ldots \vee x_n,$$

for $x = (x_1, \ldots, x_n) \in 2^X$, can be calculated by bi-quadratic cooperative Boolean circuits with $\leq 2|X|$ variables in $\leq \lceil \log(|X|) \rceil$ steps.

Proof. Group variables into pairs and use a cooperative bi-quadratic Boolean circuit of depth 1 to calculate $x_1 \wedge x_2, x_3 \wedge x_4, \ldots, x_{n-1} \wedge x_n$. If n is odd, use $x_n \wedge x_n$ for the last pair instead. Then use induction and concatenation of Boolean circuits. The total number of internal variables can be estimated as in the proof of Proposition 10.1. □

Corollary 10.3. *For every positive integer r there exist positive integers $u(r)$ and $d(r)$ such that every cooperative Boolean function $f : 2^r \to 2^r$ can be calculated by a cooperative bi-quadratic Boolean circuit with at most $u(r)$ internal variables that calculates f in $d(r)$ steps.*

Proof. Note that the Conjunctive or the Disjunctive Normal Form of cooperative Boolean functions do not use negations and use Propositions 10.1 and 10.2. □

10.2. **The proof of Lemma 5.1.** Assume the pair (k, ε) is c-friendly and $(1 + \varepsilon)\log(n) = k\ell$. Let F_1^n be a function that takes as input a pair of vectors (s_X, s_R) with $|X| = \log(n)$ and $|R| \leq \gamma_R \log(\log(n))$ for a fixed constant γ_R such that F_1^n returns the code for $V(s_X) + 1$ modulo $n - i$ whenever s_X is coding and s_R is the code for i. We need to construct a cooperative bi-quadratic Boolean circuit $B = B(X \cup R, H, Z)$ of depth $\leq \gamma \log(n)$ with $|H| \leq \gamma(\log(n))^2$ for some fixed γ that calculates F_1^n.

Since all codes are pairwise incomparable, by Proposition 2.1 we may assume that F_1^n is cooperative, and hence can be calculated by a cooperative bi-quadratic Boolean circuit. Unfortunately, we cannot use Lemma 10.3 directly, since it does not give us sufficient control on how fast the depth and number of internal variables of this circuit grow with n. Our strategy in this proof will be to break down the codes of the integers involved into small chunks of fixed size, use Lemma 10.3 to build Boolean circuits of fixed size and depth that calculate certain auxiliary functions g_1, g_2, g_3, g_4 on these chunks, and to use Propositions 10.1 and 10.2 to calculate auxiliary functions h_1, h_2 that keep track of the global picture and allow us to concatenate the smaller Boolean circuits into a larger one with the desired properties.

Let $D \subset C_k$ be such that $|D| = K = 2^{\frac{k}{1+\varepsilon}}$ and let $w : D \to \{0, \ldots, K - 1\}$ be a bijection. In this proof it will be convenient to use the notation x for the input vector s_X and the notation r for the input vector s_R of our Boolean circuit. Then $x \in 2^{k\ell}$ and wlog we may assume that $r \in 2^{km}$ for some $m < \ell$. A vector x (r) is coding if $x \in D^\ell$ $(r \in D^m)$. We partition x, r into consecutive vectors $x^j \in 2^{X^j}, r^j \in 2^{R^j}$ of length k each (as in Section 8). The corresponding decoding function V for x is then defined by

$$V(x) = \sum_{j=0}^{\ell-1} w(x_j)K^j. \tag{28}$$

Consider four functions with the following properties:

- $g_1 : C_k \to 2^2$ returns $(1, 0)$ for the input $x^j \in D$ with $w(x^j) = K - 1$ and returns $(0, 1)$ on all other inputs in D.
- $g_2 : C_k \times 2^2 \to C_k$ is such that $g_2(x^j, 1, 0), g_2(x^j, 0, 1) \in D$ with $w(g_2(x^j, 1, 0)) = w(x^j) + 1$ and $w(g_2(x^j, 0, 1)) = x^j$ whenever $x^j \in D$.
- $g_3 : C_k \times C_k \to 2^2$ is such that $g_3(x^j, x^j) = (1, 0)$ whenever $x^j \in D$, and $g_3(x^j, y^j) = (0, 1)$ whenever $x^j, y^j \in D$ with $x^j \neq y^j$.
- $g_4 : C_k \times 2^2 \to C_k$ is such that $g_4(x^j, 1, 0), g_4(x^j, 0, 1) \in D$ with $w(g_4(x^j, 1, 0)) = 0$ and $w(g_4(x^j, 0, 1)) = w(x^j)$ whenever $x^j \in D$.

Since the requirements we have specified for these functions only pertain to subsets $D, D \times \{(0, 1), (1, 0)\}$, or $D \times D$ of the domains that consist of pairwise incomparable vectors, we may assume that these functions are cooperative. Thus they can be calculated by cooperative, bi-quadratic Boolean circuits B_1, B_2, B_3, B_4 whose sizes and depths are constant in the sense that they do not depend on n.

Let $h_1 : (2^2)^\ell \to (2^2)^\ell$ be a cooperative function such that the j-th coordinate of its output is $(1,0)$ whenever the i-th coordinates of its input vector are $(1,0)$ for all $i < j$, and is $(1,0)$ whenever at least one of the i-th coordinates of its input vector is $(0,1)$ for some $i < j$. By Proposition 10.2, such h_1 can be calculated by a cooperative, bi-quadratic Boolean circuit B^1 with $\leq 4\,|\ell|$ variables in $\leq \lceil \log(|\ell|)\rceil$ steps. If we compose a product of ℓ copies of the functions g_1 on individual components x^j of x with h_1, we obtain a vector in $(2^2)^\ell$ that indicates the location of carry-over digits in the operation $w(x)+1$ as performed on the codes, with $(1,0)$ indicating that $w(x^j)+1$ needs to be calculated, whereas $(0,1)$ signals that x^j needs to be copied. This composition can be calculated by a bi-quadratic cooperative Boolean circuit B_c with $\leq \gamma_1\,|\ell|$ variables and depth $\leq \gamma_1\lceil\log(|\ell|)\rceil$, where the constant γ_1 depends on the size and depth of B_1.

If we first duplicate the value of x, keep copying it until B_c finishes its calculations, and then use the copies of x^j as the first inputs in copies of B_2, with the coordinates of the output of B_c providing the second inputs, we can construct a cooperative bi-quadratic Boolean circuit B_a with $\leq \gamma_2\,|\ell|$ variables and depth $\leq \gamma_2\lceil\log(|\ell|)\rceil$ that calculates the code for $w(x)+1$ *mod* n for every coding vector x. Here γ_2 is another constant that depends on γ_1 and the size and depth of B_2.

It remains to compare the output of B_a with $n-i$, where i is coded by r and reset it to zero if in fact the two vectors are equal. So far, we have not specified how i is coded by the elements of the counter; it will be most convenient if we assume that r codes the last mk binary digits of $n-i$ according to (28). Since $m \ll \ell$, we have a code for i in this sense iff for the partition of the output of B_a into consecutive y^js we have $y^j = r^j$ for all $j < m$ and $w(y^j) = K-1$ for $j \geq m$. We can use a product of m functions of the form g_3 with $\ell - m$ functions of the form g_1 to code the equality as a vector in $(2^2)^\ell$ all of whose coordinates are $(1,0)$, while inequality will be signified by at least one coordinate of the form $(0,1)$. Composing this product with a cooperative function $h_2 : (2^2)^\ell \to 2^2$ that returns $(1,0)$ on the code for equality and $(0,1)$ on all codes for inequality results in a function that detects a code for $n-i$. Again, by Proposition 10.2, this composition can be calculated by a cooperative, bi-quadratic Boolean circuit B_e with $\leq \gamma_3\,|\ell|$ variables and depth $\leq \gamma_3\lceil\log(|\ell|)\rceil$, where the constant γ_3 depends on the size and depth of B_3.

To put it all together, we need to copy the value of r until it will be needed for the calculation of B_e, and also retain a copy y of the output of B_a until B_e has finished its calculations. Finally, we can use the coordinates of that copy of y and of the output of B_e as inputs to ℓ copies of the Boolean circuit B_4 that calculates g_4. This adds $\leq \gamma_4\,|\ell|$ variables and $\leq \gamma_4\lceil\log(|\ell|)\rceil$ steps to the final Boolean circuit B. Since $\log(n) \leq \ell$, we conclude that B will have depth $\leq \gamma\log(n)$ and $\leq \gamma(\log(n))^2$ variables for some fixed γ.

It is straightforward to verify that B is cooperative, bi-quadratic, and calculates F_1^n as required. □

10.3. **The proof of Lemma 5.2.** Lemma 5.2 follows from Lemma 10.4 below by letting R be the set of variables for our counter and $T = |I|$, with g coding the sequence of desired values of the counter. By (26) and the choice of I we have

$|R|T \leq (log(n))^{\alpha}$ for all $\alpha > 1$ and sufficiently large n, and therefore B_c satisfies the restrictions on its depth and number of variables that are implied by Lemma 5.2.

Lemma 10.4. *Let $0 < q < 1$. Then there exists a positive constant γ such that for every nonempty set R and positive integer T and every function $g : \{0, \dots, T-1\} \to 2^R$ there exists a Boolean input-output system $B_c = B_c(\emptyset, H, R)$ with the following properties:*

(i) $|H| \leq \gamma |R| T \log(|R|T)$;

(ii) $d := d(B_c) \leq \gamma \log(|R|T)$;

(iii) B_c is cooperative and bi-quadratic;

(iv) There exists a fixed state $s^*_H \in 2^H$ such that for a proportion of $> q$ of all possible initial states of the variables in H the following two conditions hold:

(v) $s_H(2d+1) = s^*_H$;

(vi) For all times $t \geq d$ the system writes the value $g(t \ mod \ T)$ to its output variables R.

Proof. Let q, R, T, g be as in the assumption. Note that we may wlog assume that $|R|T$ is sufficiently large, since we can replace T with $T^+ = kT$, choose $g(t) = g(t-T)$ for $t \geq T$, and make γ sufficiently large to cover the finite set of instances whose original size $|R|T$ was too small for our probability estimates.

In order to make the idea of our construction more transparent, let us begin by specifying the dynamics on the highest level $H(d-1)$ and the output variables. The set $H(d-1)$ will be partitioned into pairwise disjoint subsets H_j for $j = 0, \dots, T-1$, each of size $|R|$. For any $t > d-1$, let $t^* = t - d + 1 \ mod \ T$.

Let us enumerate $H_j = \{h_{i,j} : \ i \in [|R|]\}$ for $j \in \{0, \dots, T-1\}$ and let us provisionally define

$$h_{i,j}(t+1) = h_{i,j-1}(t), \tag{29}$$

where we interpret $j - 1 = -1$ as $T - 1$. This definition will give us the correct output sequence in the sense of (vi) as long as $s_{H_j}(d-1) = g(t)$ for all $t \in \{0, \dots, T-1\}$. In other words, we need to make sure that, with sufficiently high probability, each of the variables $s_{h_{i,j}}$ assumes a certain value $h^*_{i,j}$ at time $d-1$, and that at all subsequent times the dynamics of $H(d-1)$ is identical to the dynamics specified by (29). If we can achieve this, then we can arrange for the output vector $s_R(t+1)$ to simply be a copy of $H_{T-1}(t)$.

The following trick allows us to achieve this goal. Let

$$P_0 = \{h_{i,j} \in H(d-1) : h^*_{i,j} = 0\}, \ P_1 = \{h_{i,j} \in H(d-1) : h^*_{i,j} = 1\}.$$

We will construct B_c in such a way that there exist variables $k_{i,j} \in H(d-2)$, where $H(d-2)$ is the level right below $H(d-1)$, $i \in |R|$, $j < T$ such that with probability $> q$:

$$\begin{aligned} &\forall h_{i,j} \in P_0 \ s_{k_{i,j}}(d-2) = 0 \ \& \ \forall t > d-2 \ s_{k_{i,j}}(t) = 1, \\ &\forall h_{i,j} \in P_1 \ s_{k_{i,j}}(d-2) = 1 \ \& \ \forall t > d-2 \ s_{k_{i,j}}(t) = 0. \end{aligned} \tag{30}$$

Now we modify (29) as follows:

$$\begin{aligned} &\text{If } h_{i,j} \in P_0, \text{ then } s_{h_{i,j}}(t+1) = s_{h_{i-1,j}}(t) \vee s_{k_{i,j}}(t), \\ &\text{If } h_{i,j} \in P_1, \text{ then } s_{h_{i,j}}(t+1) = s_{h_{i-1,j}}(t) \wedge s_{k_{i,j}}(t). \end{aligned} \tag{31}$$

Notice that in view of (30), our revised definition (31) of (29) ensures that with probability $> q$, at time $d-1$ all variables $h_{i,j}$ will take their desired values, and the input from level $H(d-2)$ will not influence the dynamics at level $H(d-1)$ at any time $t > d-1$.

Therefore, given $q < 1$, a set P of sufficiently large size, and a partition $P_0 \cup P_1$ of P, it suffices to construct a system $B_c(\emptyset, H, P)$ of depth d, and $s^*_H \in 2^H$ such that for some fixed $\gamma > 0$ that does not depend on $|P|$, with probability $> q$:

(a) $|H| \leq \gamma |P| \log(|P|)$;
(b) $d \leq \gamma \log(|P|)$;
(c) for $p_0 \in P_0$ we have $s_{p_0}(d) = 0$ and $s_{p_0}(t) = 1$ for all $t \geq d+1$;
(d) for $p_1 \in P_1$ we have $s_{p_1}(d) = 1$ and $s_{p_1}(t) = 0$ for all $t \geq d+1$;
(e) $s_H(2d+1) = s^*_H$,

where points (c), (d), (e) will hold with probability $> q$.

We show how to construct such $B_c(\emptyset, H, P)$, provided that P_0 is empty. If this is not the case, we can construct a dual system to take care of the case $P_1 = \emptyset$ in an analogous way and take the disjoint union of the two systems.

Let H_l be the lowest level of H, let H_h be the highest level, and let H_r denote all the remaining levels. Wlog we assume that $|P|/16$ is an integer, and that H_l and H_h consist of $K = 4\,|P|$ variables. Let

$$P = \{p(i) : i \leq K/4\}, \quad H_l = \{h_l(i) : i \leq K\}, \quad H_h = \{h_h(i) : i \leq K\}.$$

Now let

$$\{h^*(1), \ldots, h^*(K)\} \tag{32}$$

be an enumeration of the variables $h_h(1), h_h(2) \ldots, h_h(K/2)$ such that each variable in this set gets listed exactly twice. Similarly, let

$$\{h^{**}(1), \ldots, h^{**}(K/4)\} \tag{33}$$

be an enumeration of the variables $h_h(9K/16+1), \ldots, h_h(11K/16)$, such that each variable in this set gets listed exactly twice. The regulatory functions for the lowest level H_l are given by

$$s_{h_l(i)}(t+1) = s_{h_l(i)}(t) \wedge s_{h^*(i)}(t),\ i \leq K \tag{34}$$

and for the output variables by

$$s_{p_i}(t+1) = s_{p_i}(t) \wedge s_{h^{**}(i)}(t),\ i \leq K/4. \tag{35}$$

Now let H_r be an implementation of the sorting function $f_s : H_l \to H_h$, that is, a function such that the number of zeros in x is the same as in $f(x)$ but all the zeros in $f(x)$ precede all the ones in $f(x)$. By a result in [1], there exists a *sorting network* of depth $O(\log(K))$ that will implement f_s. A sorting network performs at each step a certain number of pairwise comparisons on disjoint sets of two variables and switches the variables if they are in the wrong order. In the Boolean context, this operation can be implemented as $sw(s_i, s_j) = [s_i \wedge s_j, s_i \vee s_j]$; a variable that does not participate in any comparison at a given step will simply be copied. Thus the function f_s can be calculated by a cooperative bi-quadratic Boolean circuit of depth $O(\log(K))$ with $O(K\log(K))$ variables, so the system $B_c(\emptyset, H, P)$ satisfies (a) and (b).

Let us make a few observations about the construction that we have described. First, note that we will end up with a Boolean input-output system rather than a Boolean circuit since there is feedback from levels H_h and H_l itself to level H_l. This will be the only feedback loops, the variables at all other levels of H will take input only from the next lower level.

Now define for all times $t \geq 0$:

$$S(t) = K - \sum_{i \leq K} s_{h_l(i)}(t). \tag{36}$$

Note that the self-feedback (34) at level H_l implies that

$$\forall t \; S(t+1) \geq S(t). \tag{37}$$

For a randomly chosen initial state, $S(t)$ is a random variable. Note that $S(0)$ is a binomial random variable that counts the number of failures (zeros) in K independent trials with success probability 0.5. In contrast, $S(1)$ is a binomial random variable that counts the number of failures in K independent trials with success probability 0.25. A straightforward application of the Central Limit Theorem shows that

$$Pr(S(0) < 9K/16 \;\&\; S(1) > 11K/16) > q$$

for all sufficiently large K.

This means that for a proportion of $> q$ states we will have $s_{h_h(i)}(d-1) = 1$ for $9/16 < i \leq 11/16$, and $s_{h_h(i)}(d) = 0$ for $9/16 < i \leq 11/16$, or

$$s_{p(i)}(d) = 1 \quad \text{for all} \quad i \leq 1/4K,$$
$$s_{p(i)}(d+1) = 0 \quad \text{for all} \quad i \leq 1/4K.$$

By (37), we get that

$$s_{p(i)}(t) = 0 \text{ for } i \leq K/4 \text{ and } t \geq d+1,$$

so (c) and (d) hold. Also, $s_{h_h(i)}(d) = 0$ for $i \leq K/2$, so $s_{h_l(i)}(d+1) = 0$ for $i \leq K$. By (37) again,

$$s_{h_l(i)}(t) = 0 \text{ for all } i \leq K \text{ and } t \geq d+1,$$

which clearly implies (e). □

10.4. **The proof of Lemma 5.3.** Recall the definition of crude vectors from the end of Section 8 and let $r = |R_i|$. The following technical result is identical with Lemma 5.3, except that we relabeled the input vectors for convenience as $z \in 2^{k\ell+r}$ instead of $(s_{X_{i_0}}, s_R)$, with $z \restriction 2^{k\ell}$ playing the role of $s_{X_{i_0}}$, and consider the output vector simply as an element of $2^{k\ell}$ instead of specifying X_{i_1} as its set of coordinates.

Lemma 10.5. *Fix k. For every positive integer ℓ let $\mathcal{Z}$ be the set of all vectors $z \in 2^{k\ell+r}$ that are crude or such that $x = z \restriction 2^{k\ell}$ is coding. Fix a coding vector $x^* \in 2^{k\ell}$. Then there exists an L-function $F_3 : 2^{k\ell+r} \to 2^{k\ell}$ that satisfies for all inputs $z \in \mathcal{Z}$:*

- *$F_3(z) = x = z \restriction 2^{k\ell}$ if z is coding;*
- *$F_3(z) = x^*$ if z is crude.*

Proof. Let $a = \ell + \frac{r}{k} - 1$. For $z = (z^0, \dots, z^a)$ define

$$ONE(x) = \vee(\wedge(z^0), \dots, \wedge(z^a)), \qquad ZERO(x) = \wedge(\vee(z^0), \dots, \vee(z^a)).$$

Observe that $ONE(z) = 1$ implies that z is crude or $z \notin \mathcal{Z}$ and $ZERO(z) = 0$ also implies that z is crude or $z \notin \mathcal{Z}$.

For simplicity of notation, identify x^* with the set of coordinates where this vector takes the value 1. Define the output of F_3 coordinatewise for $i < k\ell$ by

$$\begin{aligned} (F_3(z))_i &= z_i \vee ONE(x) \qquad \text{for} \qquad i \in x^* \\ (F_3(x))_i &= z_i \wedge ZERO(x) \qquad \text{for} \qquad i \notin x^*. \end{aligned} \tag{38}$$

Now, if $z \in \mathcal{Z}$ is not crude, then $x = z \restriction 2^{k\ell}$ is coding and so $x_i \vee ONE(x) = x_i$, $x_i \wedge ZERO(x) = x_i$, and $F_3(z) = x$. On the other hand, because of the choice of clauses in (38), $F_3(z) = x^*$ if z is crude.

Building a cooperative Boolean circuit of suitable depth and size that depend on ℓ and hence on n with $(1 + \varepsilon)\log(n) = k\ell$ that implements the calculations of $ONE(z)$, $ZERO(z)$, and (38) is straightforward using the technique of the proof of Lemma 5.1. □

11. Conclusion and future directions

We have constructed examples of cooperative Boolean systems with synchronous updating in which most initial conditions belong to the basin of attraction of a single attractor that can be of length $> c^N$ for any $c < 2$ and sufficiently large N. This contrasts sharply with results on various other types of dynamical systems that show nongenericity or absence of non-steady state attractors under the assumption of cooperativity. These systems are also bi-quadratic. It is shown in [7] that upper bounds on the proportion of monic regulatory functions imply nontrivial upper bounds on c for bi-quadratic cooperative Boolean systems with an attractor of length $> c^N$; in particular, if this proportion is zero, then $c < 10^{1/4}$. In the follow-up paper [9] we will re-examine this upper bound in the context of p-c-chaos.

Exponentially long attractors in Boolean systems are a hallmark of chaotic dynamics. Two other important hallmarks of chaos are a small fraction of eventually frozen nodes and high sensitivity to initial conditions. Sensitivity to initial conditions can be formalized in a variety of ways. For example, let us call a Boolean network *p-unstable* if a random single-bit flip in a randomly chosen initial condition moves the system to the basin of attraction of a different attractor with probability $> p$. The systems that we constructed in this paper are extremely chaotic with respect to attractor length and the expected proportion of eventually frozen nodes, but also p-coalescent, which contradicts $1 - p$-instability. In the follow-up paper [9] we will construct, for any given $0 < p < 1 < c < 2$, examples of Boolean systems that are simultaneously p-c-chaotic and p-unstable. However, Theorem 5 of [7] shows that for $\sqrt{3} < c$ and $p > 0.75 + \frac{\ln 0.5c}{2 \ln 0.75}$ no cooperative Boolean system that uses only binary AND and binary OR as regulatory functions can be simultaneously p-unstable and have an attractor of length ≥ 0.75. We will improve upon this result in the context of p-c-chaos and also explore the relation to other notions of sensitive dependence on initial conditions in [9].

References

[1] M. Ajtai, J. Komlos, and E. Szemeredi, *Sorting in $c \log n$ parallel steps,* Combinatorica 3(1983), pp. 1–19.

[2] M. Aldana, S. Coppersmith, and L. P. Kadanoff, *Boolean Dynamics with Random Couplings,* in *Perspectives and Problems in Nonlinear Science,* E. Kaplan, J. E. Marsden, and K. R. Sreenivasan, eds., Springer Verlag, New York, NY, 2003, pp. 23–90.

[3] Arcena, J. Demongeot, and E. Goles, *On limit cycles of monotone functions with symmetric connection graph,* Theor. Comput. Sci. 322(2004), pp. 237–244.

[4] G. Enciso, M. Hirsch, and H. Smith, *Prevalent behavior of strongly order preserving semiflows,* J. Dyn. Diff. Eq. 20(2008), pp. 115–132.

[5] G. A. Enciso and W. Just, *Analogues of the Smale and Hirsch theorems for cooperative Boolean and other discrete systems,* J. Differ. Equ. Appl., to appear.

[6] B.Fahri, *Nontrivial lower bounds for the least common multiple of some finite sequences of integers,* J. Number Theory 125(2007), pp. 393-411.

[7] W. Just and G. A. Enciso, *Extremely Chaotic Boolean Networks,* arXiv:0811.0115v1 (2008). Available at http://arxiv.org/abs/0811.0115

[8] W. Just and G. Enciso, *Exponentially long orbits in Boolean networks with exclusively positive interactions,* Nonlinear Dynamics and Systems Theory 11(2011), pp. 275–284.

[9] W. Just and M. Malicki, *Cooperative Boolean Systems with Generically Long Attractors II,* in preparation.

[10] B. Drossel, *Random Boolean Networks,* in *Reviews of nonlinear dynamics and complexity, volume 1*, H. G. Schuster, ed., Wiley-VCH, Weinheim, Germany, 2008, pp. 69–110.

[11] W. Just, (2010). *p-c-chaos is not generic for cooperative Boolean systems,* research note, 2010.

[12] M. Hirsch, *Stability and convergence in strongly monotone dynamical systems,* Reine und Angew. Math. 383(1988), pp. 1–53.

[13] S. A. Kauffman, *Homeostasis and differentiation in random genetic control networks,* Nature 224(1969), pp. 177–178.

[14] S. A. Kauffman, *Metabolic stability and epigenesis in randomly constructed genetic nets,* J. Theor. Biol. 22(1969), pp. 437–467.

[15] S. A. Kauffman, *Origins of Order: Self-Organization and Selection in Evolution,* Oxford U Press, Oxford, UK, 1993.

[16] V. Kaufman and B. Drossel, *On the properties of cycles of simple Boolean networks,* Eur. Phys. J. B 43(2005), pp. 115–124.

[17] U. Paul, V. Kaufman, and B. Drossel, *Properties of attractors of canalyzing random Boolean networks,* Phys. Rev. Lett. 73(2006), 026118, pp. 1–9.

[18] E. Remy and P. Ruet, *From minimal signed circuits to the dynamics of Boolean regulatory networks,* Bioinformatics 24(2008), pp. i220–i226.

[19] E. Remy, P. Ruet, and D. Thieffry, *Graphic requirements for multistability and attractive cycles in a Boolean dynamical framework,* Adv. Appl. Math. 41(2008), pp. 335–350.

[20] H. Smith, H. *Monotone Dynamical Systems: An Introduction to the Theory of Competitive and Cooperative Systems,* AMS, Providence, RI, 1995.

[21] E. D. Sontag, *Monotone and near-monotone biochemical networks,* Journal of Systems and Synthetic Biology 1(2007), pp. 59–87.

[22] E. D. Sontag, A. Veliz-Cuba, R. Laubenbacher, and A. S. Jarrah, *The effect of negative feedback loops on the dynamics of Boolean networks,* Biophys. J. 95(2008), pp. 518–526.

[23] D. Thieffry, *Dynamical roles of biological regulatory circuits,* Brief. Bioinform. 8(2007), pp. 220–225.

[24] R. Thomas and R. d'Ari, *Biological feedback,* CRC Press, Boca Raton, FL, 1990.

Winfried Just: Department of Mathematics, Ohio University, Athens, OH 45701, USA, Maciej Malicki: Institute of Mathematics of the Polish Academy of Sciences, Sniadeckich 8, 00-956, Warsaw, Poland

E-mail address: `mathjust@gmail.com, mamalicki@gmail.com`